UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/698,498 | 10/30/2003 | Sanjay Aiyagari | 50325-0805 | 9591 |

29989 7590 09/21/2006

HICKMAN PALERMO TRUONG & BECKER, LLP
2055 GATEWAY PLACE
SUITE 550
SAN JOSE, CA 95110

| EXAMINER | |
|---|---|
| KIM, PAUL | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2161 | |

DATE MAILED: 09/21/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/698,498 | AIYAGARI ET AL. | |
| | Examiner | Art Unit | |
| | Paul Kim | 2161 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *04 August 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-21 and 39-59* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21 and 39-59* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *30 October 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

SAM RIMELL
PRIMARY EXAMINER

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

## DETAILED ACTION

1. This Office action is responsive to the following communication: Amendment filed on 4 August 2006.

### *Response to Amendment*

2. Claims 1-21 and 39-59 are pending and present for examination. Claims 1, 18, 39, 48 and 56 are independent.

3. Claims 22-38 have been cancelled.

4. Claims 39-59 have been added.

5. Claims 1, 6 and 10 have been amended.

### *Drawings*

6. As per the objection to the Drawings, applicant's amendment has been acknowledged. Accordingly, the objection has been withdrawn.

### *Specification*

7. As per the objection to the Abstract, applicant's amendment has been acknowledged. Accordingly, the objection has been withdrawn.

### *Claim Rejections - 35 USC § 102*

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. **Claims 1-21 and 39-59** are rejected under 35 U.S.C. 102(b) as being anticipated by Barkley et al (U.S. Patent No. 6,202,066, hereinafter referred to as BARKELY), filed on 18 November 1998, and issued on 13 March 2001.

10. **As per claims 1, 10, 18, 22, 31, 39, 48 and 56**, BARKLEY teaches:

A method for controlling access to a resource, the method comprising the steps of:

creating and storing in ~~the Operating System~~ a filesystem of an Operating System a file that represents the resource {See BARKLEY, col. 2, lines 23-26, wherein this reads over "object security attributes are usually kept with the object (e.g., in the header of a file) and the object resides in (or a resource is accessed through) a single server"; and col. 4, lines 59-60, wherein this reads over "OATS can be created, edited, deleted, and assigned to or removed from objects. Each OAT thus defines an access control specification"};

receiving user-identifying information from a user requesting access to the resource {See BARKELY, col. 1, lines 48-54, wherein this reads over "Windows NT allows various permission to be associated by the ACL with individuals or groups of individuals, so that the access sought is permitted"}, wherein the user-identifying information comprises a role associated with the user, wherein the role is determined from a user identifier uniquely associated with the user and from a group identifier associated with a group that includes the user {See BARKLEY, col. 2, lines 4-14, wherein this reads over "[u]ser security attributes may consist of defined groups ('roles') to which the user belongs, wherein access to various objects is permitted to all of the individuals identified as members of the group"};

receiving a resource identifier associated with the resource {See BARKLEY, col. 1, lines 22-27, wherein this reads over "'objects' may also include resources"; and col. 7, lines 22-26, wherein this reads over "a mechanism for mapping permissions authorized with respect to various objects to the corresponding identified individuals or groups"};

creating an access identifier based on the user-identifying information and the resource identifier, wherein the access identifier is formatted as a file attribute that is used by the Operating System to manage file access {See BARKLEY, col. 2, lines 23-26, wherein this reads over "object security attributes are usually kept with the object (e.g., in the header of a file) and the object resides in (or a resource is accessed through) a single server"; and col. 4, lines 59-60, wherein this reads over "OATS can be created, edited, deleted, and assigned to or removed from objects. Each OAT thus defines an access control specification"};

calling the Operating System to perform an operation on the file using the access identifier to gain access to the file {See BARKLEY, col. 8, lines 25-38, wherein this reads over "In the Windows NT implementation mentioned . . . OATS [Object Access Type] which can be associated with objects, and writes the permissions and users (or roles) associated with each objects to the access control lists" and "access to a given object permitted only if an OAT assigned to that object itself indicated that the requestor was a member of a role having been assigned the permission sought with respect to the object"}; and

granting the user access to the resource only if the Operating System call successfully performs the operation {See BARKLEY, col. 8, lines 25-38, wherein this reads over "access to a given object permitted only if an OAT assigned to that object itself indicated that the requestor was a member of a role having been assigned the permission sought with respect to the object"}.

11. **As per dependent claims 2, 11, 19, 23, 32, 40, 49 and 57**, BARKLEY teaches:

A method as recited in Claim 1, wherein the access identifier comprises:

a first set of bits for storing a role identifier, wherein the role identifier is associated with the role {See BARKLEY, col. 2, lines 4-14, wherein this reads over "user security attributes may consists of defined groups ('roles')"; lines 23-26, wherein this reads over "object security attributes are usually kept with the object (e.g., in the header of a file) and the object resides in (or a resource is accessed through) a single server"; and col. 4, line 47 – col. 5, line 4, wherein this reads over "Object Access Type" and "adding that role, assigned to those users, to the corresponding OAT"}; and

a second set of bits for storing the resource identifier {See BARKLEY, col. 4, lines 59-60, wherein this reads over "OATS can be created, edited, deleted, and assigned to or removed from objects. Each OAT thus defines an access control specification"}.

Furthermore, it would be inherent store both a role identifier and a resource identifier on a set of bits such that the set(s) of bits may be called and received by an Operating System as described in the aforementioned method of Claim 1.

12. **As per dependent claims 3, 20, 24, 41 and 58**, BARKLEY teaches:

A method as recited in Claim 1, wherein:

the step of creating an access identifier based on the user-identifying information and the resource identifier comprises formatting the access identifier as a group identifier file attribute {See BARKLEY, col. 7, lines 14-20, wherein this reads over "a simple mechanism for thus associating groups of object with sets of permissions and of users, organized as roles or groups"}; and

the step of calling the Operating System to perform an operation on the file representing the resource comprises:

assigning the access identifier to a group identifier attribute of an Operating System process {See BARKLEY, col. 9, lines 1-7, wherein this reads over "allowing a system administrator to add or remove a role or group from the OAT"}; and

calling an Operating System routine from the Operating System process to perform the operation on the file representing the resource {See BARKLEY, col. 8, lines 25-38, wherein this reads over "In the Windows NT implementation mentioned . . . OATS [Object Access Type] which can be associated with objects, and writes the permissions and users (or roles) associated with each objects to the access control lists" and "access to a given object permitted only if an OAT assigned to that object itself indicated that the requestor was a member of a role having been assigned the permission sought with respect to the object"}.

13. **As per dependent claims 4, 13, 25, 34, 42 and 51**, BARKLEY teaches:

A method as recited in Claim 1,

wherein the step of calling the Operating System to perform an operation on the file representing the resource comprises comparing the access identifier to an identifier included in an Access Control List file attribute associated with the file representing the resource {See BARKLEY, col. 1, lines 48-54, wherein this reads over "access sought is permitted only if the user's identification matches the user entry in the ACL or the user is a member of a group entry in the ACL"},

wherein the Access Control List file attribute includes the identifiers of all users and all groups of users allowed to access the file representing the resource {See BARKLEY, col. 1, lines 48-54, wherein this reads over "a user entry in the ACL or the user is a member of a group entry in the ACL"}.

14. **As per dependent claims 5, 14, 21, 26, 35, 43, 52 and 59**, BARKLEY teaches:

A method as recited in Claim 1, wherein the operation on the file representing the resource is selected from a group consisting of opening the file, closing the file, deleting the file, reading from the file, writing to the file, executing the file, appending to the file, reading a file attribute, and writing a file attribute {See BARKLEY, col. 10, lines 56-61, wherein this reads over "possible Permissions are the usual NTFS file permissions: Read(R), Write(W), Execute(X), Delete(D), Change Permissions(P), and Take Ownership(O)"}.

15. **As per dependent claims 6, 15, 27, 36, 44 and 53**, BARKLEY teaches:

A method as recited in Claim 1, the method further comprising the steps of:

reading a permission bit associated with the file representing the resource, wherein the permission bit corresponds to ~~a file~~ the operation performable on the file representing the resource {See BARKLEY, col. 1, lines 48-54, wherein this reads over "access sought is permitted only if the user's identification matches the user entry in the ACL or the user is a member of a group entry in the ACL"};

based on the ~~file~~ operation on the file indicated by the permission bit, determining a resource operation that is performable on the resource {See BARKLEY, Figures 2, 4-5; and col. 9, lines 29-32, wherein this reads over "in green, if the selected role or group has all of the selected permissions"}; and

granting the user the privilege of performing the resource operation on the resource only if the permission bit allows the ~~file~~ operation to be performed on the file representing the resource {See BARKLEY, col. 8, lines 25-38, wherein this reads over "access to a given object permitted only if an OAT assigned to that object itself indicated that the requestor was a member of a role having been assigned the permission sought with respect to the object"}.

16. **As per dependent claims 7, 16, 28, 37, 45 and 54**, BARKLEY teaches:

A method as recited in Claim 1, the method further comprising the steps of:

opening the file representing the resource {See BARKLEY, col. 8, lines 31-38, wherein this reads over "access to a given object permitted only if an OAT assigned to that object itself indicated"};

reading from the file representing the resource a permission indicator associated with a resource operation {See BARKLEY, Fig 5; and col. 13, lines 28-49, wherein this reads over "allows a determination of permission provided by a role's membership in a hierarchy"}; and

enabling the user to perform the resource operation on the resource only if the permission indicator indicates that the user is allowed to perform the resource operation on the resource {See BARKLEY, Table 1; and col. 11, line 39 – col. 12, line 37, wherein this reads over "[v]arious roles have varied permission with respect to these files" and "only members of branch_manager can delete these files"}.

17. **As per dependent claims 8, 17, 29, 38, 46 and 55**, BARKLEY teaches:

A method as recited in Claim 1, wherein the step of representing the resource by a file stored in the Operating System filesystem comprises:

creating the file representing the resource in the Operating System filesystem {See BARKLEY, col. 2, lines 23-26, wherein this reads over "object security attributes are usually kept with the object (e.g., in the header of a file) and the object resides in (or a resource is accessed through) a single server"; and col. 4, lines 59-60, wherein this reads over "OATS can be created, edited, deleted, and assigned to or removed from objects. Each OAT thus defines an access control specification"}; and

assigning an access value to a file attribute of the file representing the resource, the file attribute being used by the Operating System to manage file access, wherein the access value corresponds to a combination of a role and a resource {See BARKLEY, col. 10, lines 56-61, wherein this reads over "possible Permissions are the usual NTFS file permissions: Read(R), Write(W), Execute(X), Delete(D), Change Permissions(P), and Take Ownership(O)"}.

18. **As per dependent claims 9, 30. 47**, BARKLEY teaches:

A method as recited in Claim 8, wherein the file attribute used by the Operating System to manage file access is a group identifier file attribute {See BARKLEY, col. 9, lines 1-7, wherein this reads over "modify the permissions associated with that role or group, to assign objects to OAT designations or remove OATs from objects"}.

19. **As per dependent claims 12, 33, and 50**, BARKLEY teaches:

A method as recited in Claim 10, wherein the step of making an Operating System call to perform an operation on the file representing the resource comprises:

storing the group identifier value of a group identifier attribute of an Operating System process {See BARKLEY, col. 2, lines 4-14, wherein this reads over "[u]ser security attributes may consist of defined groups ('roles') to which the user belongs, wherein access to various objects is permitted to all of the individuals identified as members of the group"};

assigning the access identifier to the group identifier attribute of the Operating System process {See BARKLEY, col. 2, lines 23-26, wherein this reads over "object security attributes are usually kept with the object (e.g., in the header of a file) and the object resides in (or a resource is accessed through) a single server"; and col. 4, lines 59-60, wherein this reads over "OATS can be created, edited, deleted, and assigned to or removed from objects. Each OAT thus defines an access control specification"};

calling an Operating System routine from the Operating System process to perform the operation on the file representing the resource {See BARKLEY, col. 8, lines 25-38, wherein this reads over "In the Windows NT implementation mentioned . . . OATS [Object Access

Type] which can be associated with objects, and writes the permissions and users (or roles) associated with each objects to the access control lists" and "access to a given object permitted only if an OAT assigned to that object itself indicated that the requestor was a member of a role having been assigned the permission sought with respect to the object"}, wherein the operation on the file representing the resource is performed only if the value of the group identifier attribute of the Operating System process matches the value of the group identifier file attribute of the file representing the resource {See BARKLEY, Table 1; and col. 11, line 39 – col. 12, line 37, wherein this reads over "[v]arious roles have varied permission with respect to these files" and "only members of branch_manager can delete these files"}; and

resetting the group identifier attribute of the Operating System process to the stored group identifier value {See BARKLEY, col. 9, lines 1-7, wherein this reads over "allowing a system administrator to add or remove a role or group from the OAT"}.

***Response to Arguments***

20. Applicant's arguments filed 4 August 2006 have been fully considered but they are not persuasive.

**a. Applicant's Arguments:**

i. Claim 1 under 35 U.S.C. § 102

As per independent claim 1, Applicant asserts the argument that "[t]he entire reference of Barkley fails to teach or suggest that an Operation System is called to perform an operation on a file" (See Amendment, page 20).

Secondly, Applicant asserts the argument that "Barkley fails to disclose 'creating and storing in a filesystem of an Operating System a file that represents a resource'" (See Amendment, page 21).

Thirdly, Applicant asserts the argument that Barkley fails to show at least one of "receiving a resource identifier associated with a resource" and "creating an access identifier based on the user-identifying information and the resource identifier" (See Amendment, page 21).

Lastly, Applicant asserts the argument that "Barkley does not even mention how an OAT is formatted, much less that an OAT is formatted (See Amendment, page 22).

ii. Claims 10, 18, 39, 48 and 56 under 35 U.S.C. § 102

Applicant asserts that for the reasons set forth above in connection with Claim 1, that Claims 10, 18, 39, 48 and 56 are patentable over Barkley under 35 U.S.C. § 102 (See Amendment, page 22-23).

iii. Dependent claims

Applicant asserts that each of the dependent claims is therefore allowable for the reasons given above for the claim on which it depends (Amendment, page 23).

**b. Response to Arguments:**

i. Claim 1 under 35 U.S.C. § 102

Regarding Applicant's argument that "[t]he entire reference of Barkley fails to teach or suggest that an Operating System is called to perform an operation on a file," Applicant is directed to the Background of Barkley, which states:

> "Such networks are commonly operated under control of an "operating system", which may include the capability to provide varying individuals with varying "permissions" with respect to objects stored on the file server. For example, Microsoft Corporation's "Windows NT" operating system provides this capability, by associating an "access control list" ("ACL") (this being an example of an "access control specification", as the latter term is used in the art) with each "object", e.g., with each controlled file or group of files, i.e., with a directory of controlled files. Windows NT allows various permissions to be associated by the ACL with individuals or groups of individuals, so that the access sought is permitted only if the user's identification matches the a user entry in the ACL or the user is a member of a group entry in the ACL, and the user or group entry is associated with permissions for the access sought" {See BARKLEY, C2:L35-54}.

Therefore, one of ordinary skill in the art at the time the invention was made would be able to correlate the above excerpt to specifically disclose a method wherein an Operating System, such as a Windows NT operating system, would be called to perform an operation on the file using the access identifier to gain access to the file. That is, the Windows NT operating system would grant access to files only to those individuals or groups wherein "the user's identification matches the user entry in the ACL" and "the user or group entry is associated with permissions for the access control" {See BARKLEY, C2:L48-54}. While Applicant further argues that "Barkley does not even state that an operation is performed on the file," one of ordinary skill in the art would be able to understand that it would be inherent for the Operating System to make a "call" or "open" operation on the file that represents the resource, by using the access

identifier before granting the user access to the resource. Therefore, Barkley does teach, disclose, and suggest a method wherein "an Operating System is called to perform an operation on a file."

Secondly, regarding Applicant's argument that Barkley fails to disclose "creating and storing in a filesystem of an Operating System a file that represents a resource," Applicant is directed to the Background of Barkley, which states:

> "Such networks are commonly operated under control of an "operating system", which may include the capability to provide varying individuals with varying "permissions" with respect to objects stored on the file server. For example, Microsoft Corporation's "Windows NT" operating system provides this capability, by associating an "access control list" ("ACL") (this being an example of an "access control specification", as the latter term is used in the art) with each "object", e.g., with each controlled file or group of files, i.e., with a directory of controlled files. Windows NT allows various permissions to be associated by the ACL with individuals or groups of individuals, so that the access sought is permitted only if the user's identification matches the a user entry in the ACL or the user is a member of a group entry in the ACL, and the user or group entry is associated with permissions for the access sought" {See BARKLEY, C2:L35-54}.

Therefore, one of ordinary skill in the art at the time the invention was made would be able to correlate the above excerpt to specifically disclose a method wherein the "access control list" (hereinafter referred to as "ACL") would be associated with an object or "with each controlled file or group of files, i.e., with a directory of controlled files" {See BARKLEY, C2:L42-47}. That is, Barkley specifically discloses a "controlled file or group of files" which one of ordinary skill in the art could correlate with a file that represents a resource. Therefore, Barkley does teach, disclose, and suggest a method wherein a file that represents a resource is created and stored in a filesystem of an Operating System.

Thirdly, regarding Applicant's argument that Barkley fails to show at least one of "receiving a resource identifier associated with a resource" and "creating an access identifier based on the user-identifying information and the resource identifier," it is noted that Barkley discloses the following:

> "In each of the typical types of computer systems discussed above, object security attributes are usually kept with the object (e.g., in the header of a file) and the object resides in (or a resource is accessed through) a single server. Consequently, when an object is accessed, its security attributes can be conveniently evaluated once the object has been located. Furthermore, changes in object security attributes--e.g., to add or subtract an individual from those having access of a specified type to a particular object--need only be made at a single location." {See BARKLEY, C2:L23-32}.

Therefore, one of ordinary skill in the art at the time the invention was made would be able to correlate the above excerpt to specifically disclose a method wherein the "resource identifier associated with a resource" may be correlated to the name of the file or object, and "creating an access identifier based on the user-identifying information and the resource identifier" may be correlated to the "object security attributes [which] are usually kept with the object (e.g., in the header of a file)." Therefore, Barkley does teach, disclose, and suggest a method wherein a resource identifier associated with a resource is received and an access identifier is created.

Lastly, regarding Applicant's argument that "Barkley does not even mention how an OAT is formatted, much less that an OAT is formatted," Applicant is directed to the above-mentioned excerpt of Barkley, column 2, lines 23-32, wherein the prior art discloses a method of storing "object security attributes" with the object in the header of the file.

Accordingly, the rejection of claim 1 under 35 U.S.C. 102(b) is sustained.

ii. Claims 10, 18, 39, 48 and 56 under 35 U.S.C. § 102

As per claims 10, 18, 39, 48 and 56, Applicant has not asserted any specific arguments in response to the rejections of the claims. Therefore, the rejections of claims 10, 18, 39, 48 and 56 are sustained because Applicant has not traversed the rejections nor presented any specific arguments for overcoming the rejections contained in the prior Office Action, dated 2 May 2006. Furthermore, by virtue of dependency, the rejections of Claims 10, 18, 39, 48 and 56 are sustained for the reasons stated above in relation to Claim 1.

iii. Dependent claims

As per claims 2-9, 11-17, 19-21, 40-47, 49-55, and 57-59, Applicant has not asserted any specific arguments in response to the rejections of the claims. Therefore, the rejections of claims 2-9, 11-17, 19-21, 40-47, 49-55, and 57-59 are sustained because Applicant has not traversed the rejections nor presented any arguments for overcoming the rejections contained in the prior

Office Action, dated 2 May 2006. Furthermore, by virtue of dependency, the rejections of Claims 2-9, 11-17, 19-21, 40-47, 49-55, and 57-59 are sustained for the reasons stated above in relation to Claims 1, 10, 18, 39, 48 and 56.

***Conclusion***

21. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

22. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Paul Kim whose telephone number is (571) 272-2737. The examiner can normally be reached on M-F, 9am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Christian Chase can be reached on (571) 272-4190. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Paul Kim
Patent Examiner, Art Unit 2161
TECH Center 2100

SAM RIMELL
PRIMARY EXAMINER